Verwaltung von Ferienhäusern

Aufgaben

Zur Verwaltung und Buchung von Ferienhäusern sollen ein Softwaresystem und eine Datenbank entwickelt werden.

1 Objektorientierte Entwicklung einer Anwendung
Es soll ein Buchungssystem für Ferienhäuser entwickelt werden. Sowohl Vermieterinnen oder Vermieter (im Folgenden Vermieter genannt) als auch Kundinnen oder Kunden (im Folgenden Kunde genannt) können mit dem Buchungssystem arbeiten, nachdem sie von der Administratorin bzw. dem Administrator (im Folgenden Administrator genannt) registriert wurden. Ein Ferienhaus kann abhängig vom Monat unterschiedliche Mietpreise pro Tag haben.

1.1 Das Buchungssystem soll folgende Anforderungen erfüllen.
Jeder Vermieter kann

- sich beim System anmelden,
- ein Ferienhaus zur Vermietung einstellen,
- die Preise eines Ferienhauses ändern und
- sich die Belegung eines Ferienhauses ansehen.

Ein Kunde, der ein Ferienhaus mieten möchte, kann

- sich beim System anmelden,
- ein Ferienhaus ohne Suche buchen,
- eine Buchung stornieren und
- nach Ferienhäusern suchen. Wenn er ein passendes Objekt gefunden hat, kann er es sofort buchen. Bei der Buchung wird ihm auch die Belegung angezeigt.

Ein Administrator kann

- sowohl die Rolle eines Vermieters als auch die eines Kunden annehmen
- Benutzer (Kunden oder Vermieter) beim System registrieren

Entwickeln und zeichnen Sie für die Anwendung ein Anwendungsfalldiagramm in UML-Notation.

(8 BE)

1.2 Ein erstes UML-Klassendiagramm finden Sie in Material 1.

1.2.1 Beschreiben Sie die Klasse `Ferienhaus` und ihre Beziehungen unter Berücksichtigung der Begriffe Attribut, Konstruktor, Methode, Assoziation, Navigierbarkeit, Rolle und Multiplizität.

(4 BE)

1.2.2 Überführen Sie die im UML-Klassendiagramm dargestellte Klasse `Ferienhaus` in Anweisungen einer objektorientierten Programmiersprache und implementieren Sie die aufgeführten Methoden mit Ausnahme von `ermittlePreis()`.

Hinweise: Die Liste der Buchungen eines Ferienhauses ist aufsteigend nach dem Buchungsbeginn sortiert. Die Methode `hinzufuegenBuchung()` sucht zunächst die richtige Position und fügt dann die Buchung an der ermittelten Position ein. Die Dokumentationen der Klassen `List` und `Date` sind in Material 2 zu finden.

(8 BE)

1.3 Die Methode `ermittlePreis()` der Klasse `Ferienhaus` ermittelt den Gesamt-Mietpreis für den übergebenen Zeitraum. Dabei kann es vorkommen, dass Zeiträume von mehreren Monaten für Buchungen angegeben werden, z. B. Buchung von 01.10.2023 bis 29.01.2024. Je nach Monat ist der Mietpreis dabei unterschiedlich.

Hinweis: Buchungen, die sich über mehr als ein halbes Jahr erstrecken, sind nicht möglich. Dies muss nicht überprüft werden.

1.3.1 Entwickeln Sie den Algorithmus der Methode und überführen Sie ihn in ein Struktogramm oder in Pseudocode.

(10 BE)

1.3.2 Überführen Sie die Anforderungsbeschreibung in Aufgabe 1.3 bzw. Ihre Lösung aus Aufgabe 1.3.1 in eine objektorientierte Programmiersprache.

(6 BE)

1.4 Die Methode `sucheFerienhaus()` der Klasse `Verwaltung` sucht für den übergebenen Zeitraum nach freien Ferienhäusern in einer Region, die Platz für alle Personen bieten. Alle passenden Objekte werden gesammelt und in Form einer neuen Liste zurückgegeben. Entwickeln und zeichnen Sie ein UML-Sequenzdiagramm für diese Methode.

Hinweis: Die Vorlage in Material 3 ist zu benutzen.

(10 BE)

1.5 Das Klassenmodell des Buchungssystems soll erweitert werden, damit alle Anwendungsfälle (siehe UML-Anwendungsfälle aus Aufgabe 1.1) unterstützt werden.

1.5.1 Folgende Anforderungen liegen vor:

- Personen werden unterschieden in Vermieter und Kunden, zeichnen sich durch Name, Vorname und Adresse aus und werden über eine eindeutige Nummer identifiziert.
- Vermieter besitzen zusätzlich eine IBAN, Kunden können Haustiere besitzen oder Raucher sein. Nach dem Anlegen hat ein Kunde zunächst keine Haustiere und ist Nichtraucher.
- Bei einigen Ferienhäusern ist das Rauchen bzw. das Mitbringen von Haustieren erlaubt, bei anderen nicht.
- Die Verwaltung kennt alle Vermieter.
- Ein Vermieter stellt mindestens ein Ferienhaus für eine Region ein. Zu einem Ferienhaus gibt es genau einen Vermieter. Der Vermieter kennt seine Ferienhäuser.
- Die Klasse `Ferienhaus` soll in der Lage sein, die Belegung für jeweils ein Jahr zu ermitteln. Sie wird als zweidimensionales Array vom Typ `boolean` (Monat und Tag) zurückgegeben.
- Die Preise für ein Ferienhaus (Attribut `tagesPreise[]`) können geändert werden.
- Sowohl neue Kunden als auch Vermieter können sich vom Administrator über die Klasse `Verwaltung` registrieren lassen. Danach können sie sich über die Verwaltung anmelden.
- Kunden können Buchungen stornieren, wobei sie die Buchungsnummer angeben müssen. Sie bekommen eine Rückmeldung, ob die Stornierung erfolgreich war.

Erweitern Sie die Vorlage (Material 6) um die genannten Anforderungen.

Hinweis: Die Darstellung einer Klasse `Administrator` ist nicht erforderlich. Vorhandene Klassen, Attribute und Methode müssen nicht erneut gezeichnet bzw. aufgeführt werden.

(8 BE)

1.5.2 Implementieren Sie die Klassen `Person` und `Vermieter` mit Konstruktoren ohne weitere Methoden.

(6 BE)

2 Datenbank für die Ferienhausverwaltung
Die Belegung der Ferienhäuser wird bereits in einer Datenbank verwaltet. Die in Material 4 dargestellten Relationen geben die Datenbank ausschnittsweise wieder.

2.1 Beschreiben Sie die Regeln beim Normalisierungsprozess bis zur 3. Normalform (NF). Prüfen Sie, ob die Relationen in Material 4 den Normalformen (1., 2. oder 3. NF) genügen.

(6 BE)

2.2 Beschreiben Sie drei eventuell auftretende Anomalien anhand von Beispielen aus Material 4.

(3 BE)

2.3 Überführen Sie die Tabellen aus Material 4 in das relationale Modell in der 3. Normalform unter Berücksichtigung Ihrer Ergebnisse aus den vorangegangenen Aufgaben.

Hinweis: Die Relationen sind in der Schreibweise `Relation(PK, Attribut, ... , FK#)` anzugeben.

(6 BE)

2.4 Einige Daten der Datenbank sollen ergänzt und ausgewertet werden.

2.4.1 Klaus Dahl, wohnhaft Am Lauberg 17 in 36037 Fulda, möchte sein Ferienhaus „Villa Waldesruh“ in der Rhön (Forstweg 27, 36145 Hofbieber) zur Vermietung anbieten. Das Ferienhaus bietet Platz für sechs Personen.
Formulieren Sie die SQL-Anweisungen, um die Person Klaus Dahl und das Ferienhaus in die Datenbank aufzunehmen.

Hinweis: Die Personennummer und die Objektnummer werden automatisch vergeben (`AUTO_INCREMENT`). Alle Namen sind eindeutig.

(4 BE)

2.4.2 Der Vermieter möchte eine Übersicht über die aktuellen und zukünftigen Belegungen seines Ferienhauses mit der Objektnummer 100. Die Ausgabe soll die Personennummer der Belegung, den Zeitraum und die Bezeichnung des jeweiligen Belegungstyps enthalten und nach Datum aufsteigend sortiert sein.
Entwickeln Sie eine entsprechende SQL-Anweisung.

(3 BE)

2.4.3 Die Anzahl von Ferienhäusern je Land und Regionen soll ausgegeben werden. Die Ausgabe soll nach den Ländernamen und Regionen aufsteigend sortiert sein.
Entwickeln Sie eine entsprechende SQL-Anweisung.

(3 BE)

2.4.4 Es werden die Ferienhäuser im Harz (Region-Nr. 1) für vier bis sechs Personen gesucht, die im Zeitraum vom 11.05.2024 bis 18.05.2024 nicht belegt sind.
Implementieren Sie eine entsprechende SQL-Anweisung.

(4 BE)

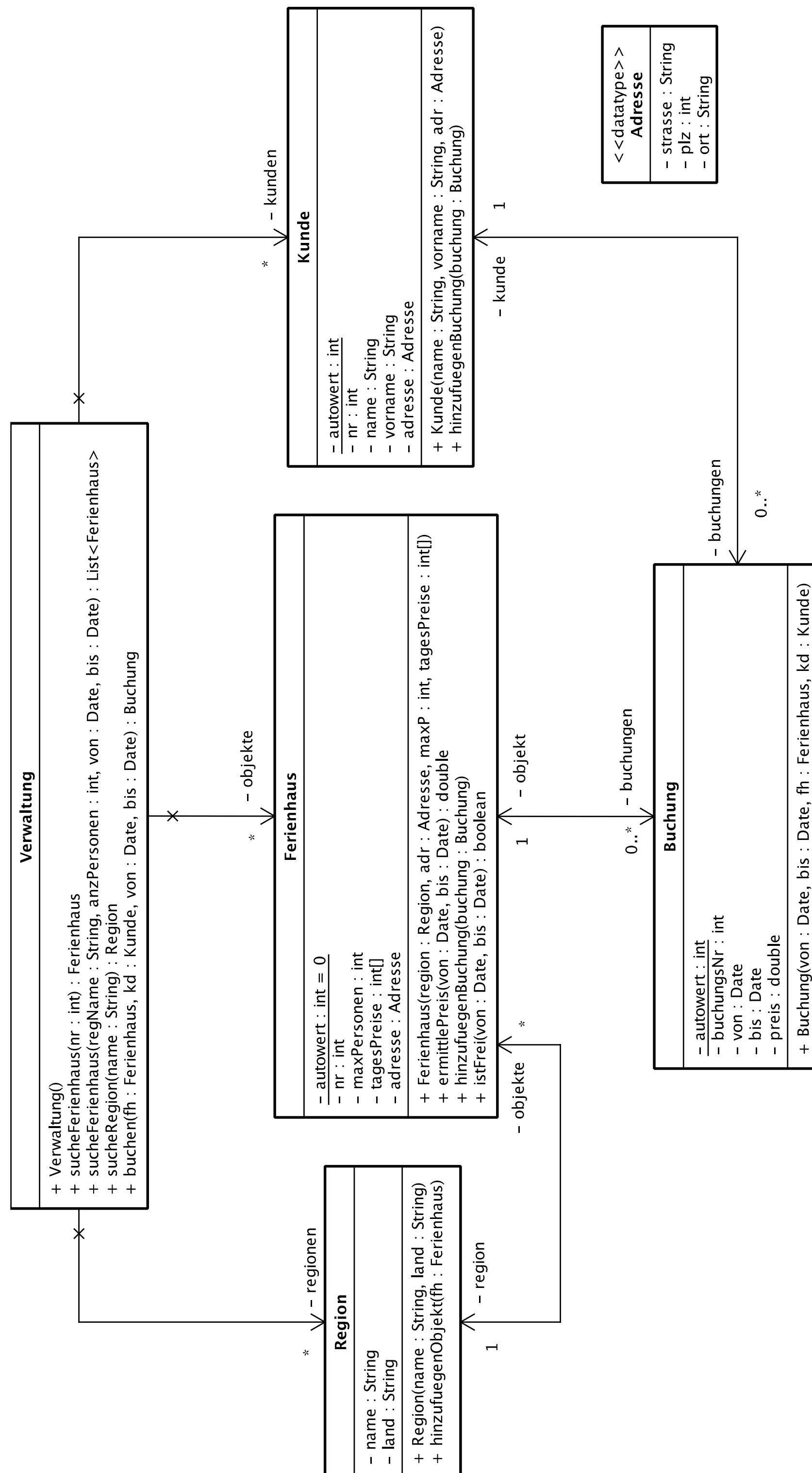
2.5 Im Zuge einer Ferienhausvermietung werden weitere Daten erfasst, die zurzeit ausschließlich in entsprechende Formulare (Textverarbeitungsdokumente) eingetragen werden. In Material 5 ist ein gekürztes, ausgefülltes Beispiel für ein Übergabeprotokoll zu einer Vermietung dargestellt. Zukünftig sollen die im Protokoll festgehaltenen Daten zu den Räumen des Ferienhauses inklusive Inventar, zum Energieverbrauch und der Schlüsselverwaltung dauerhaft in der Datenbank gespeichert werden.
Modellieren Sie auf Grundlage des Übergabeprotokolls ein ER-Modell und zeichnen Sie das Diagramm mit Entitätstypen, Attributen und Beziehungen mit Kardinalitäten in der [min, max]-Notation.

Hinweise: Räume innerhalb eines Ferienhauses haben eindeutige Raumnummern. Angaben zur Vermietung und zu Mängeln sind nicht zu berücksichtigen. Beim Inventar ist zwischen Mobiliar und Elektrogeräten zu unterscheiden. In Material 7 ist eine Vorlage zu finden.

(11 BE)

Material 1

UML-Klassendiagramm



Hinweise: Das Attribut `tagesPreise` der Klasse Ferienhaus enthält die Mietpreise je Tag abhängig vom Monat, z.B. enthält `tagesPreise[0]` den Tagesmietpreis für den Monat Januar. Falls eine Buchung nicht durchgeführt werden kann, liefert die Methode `buchen()` der Klasse `Verwaltung` den Wert `null` als Rückgabewert. Alle Attribute können mittels get-Methoden gelesen werden.

Material 2

Klassendokumentationen

Date
- tag : int - monat : int - jahr : int
+ Date() + Date(s : String) + addDays(days : int) : Date + addMonths(months : int) : Date + getNumberOfDaysInMonth() : int + equals(dt : Date) : boolean + isBefore(dt : Date) : boolean + isAfter(dt : Date) : boolean + isBetween(dt : Date, dt2 : Date) : boolean + toString() : String

Klasse `Date`

`Date()`
erzeugt ein `Date`-Objekt mit dem aktuellen SystemDate und der Uhrzeit, im Format dd.mm.yyyy.

`Date(s: String)`
erzeugt ein `Date`-Objekt auf Basis des übergebenen String-Parameters im Format dd.mm.yyyy.

`addDays(days: int): Date`
liefert eine Kopie des `Date`-Objekts, zu dem die Anzahl Tage `days` hinzuaddiert wurde.

`addMonths(months: int): Date`
liefert eine Kopie des `Date`-Objekts, zu dem die Anzahl Monate `months` hinzuaddiert wurde. Falls es den Tag im neuen Monat nicht gibt, wird der letzte Tag des Monats gesetzt (30.02.2023 → 28.02.2023).

`getNumberOfDaysInMonth(): int`
liefert die Anzahl der Tage für den Monat des Date-Objektes.

`equals(date: Date): boolean`
liefert `true`, wenn das Date-Objekt gleich dem des Parameters `date` ist.

`isBefore(date: Date): boolean`
liefert `true`, wenn das `Date`-Objekt vor dem Parameter `date` liegt.

`isAfter(date: Date): boolean`
liefert `true`, wenn das `Date`-Objekt nach dem Parameter `date` liegt.

`isBetween(dt1: Date, dt2: Date): boolean`
liefert `true`, wenn das `Date`-Objekt im Zeitraum der Parameterwerte `dt1` und `dt2` liegt.

`toString(): String`
liefert eine String-Repräsentanz des `Date`-Objekts im Format dd.mm.yyyy.

Hinweis: Alle Attribute können mittels get-Methoden gelesen werden.

Material 2 (Fortsetzung)

Klasse `List`

List<T>
+ List<T>() + add(obj : T) + add(index : int, obj : T) + contains(obj : T) : boolean + get(index : int) : T + remove(index : int) : T + remove(obj : T) : boolean + size() : int

`List<T>()`
erzeugt eine generische Liste mit Elementen des Typs `T`.

`add(obj: T)`
hängt das Objekt `obj` vom Typ `T` am Ende der Liste an.

`add(index: int, obj: T)`
fügt das Objekt `obj` vom Typ `T` an der Position `index` in die Liste ein.

`contains(obj: T): boolean`
liefert `true`, wenn das Objekt `obj` in der Liste enthalten ist, ansonsten `false`.

`get(index: int): T`
liefert das Listenelement an der Position `index` zurück bzw. `null`, falls `index` negativ oder größer gleich der Anzahl der momentan enthaltenen Elemente ist.

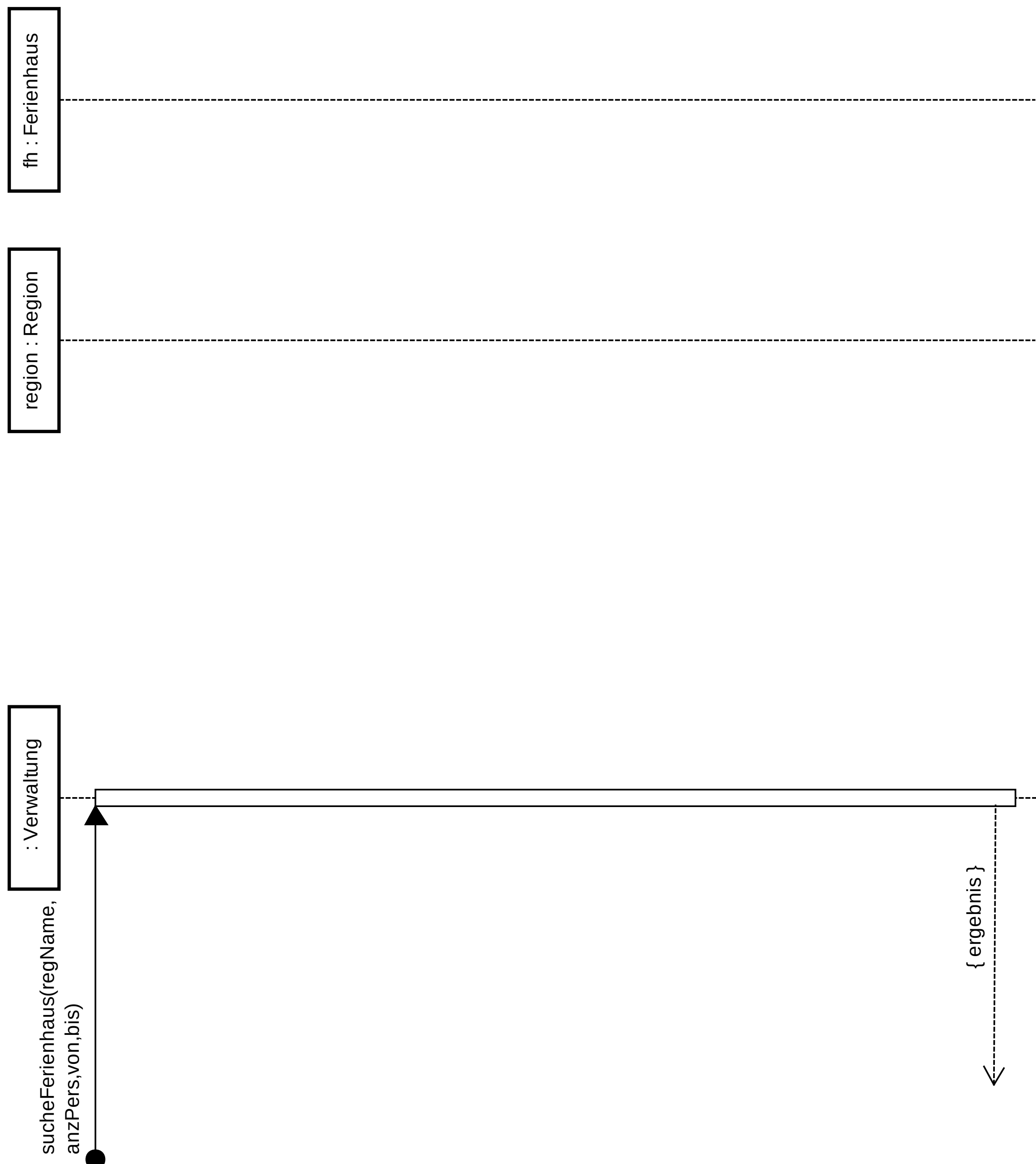
`remove(index: int): T`
entfernt das Listenelement an der Position `index`. Liefert das entfernte Element zurück bzw. `null`, falls `index` negativ oder größer gleich der Anzahl der momentan enthaltenen Elemente ist.

`remove(obj: T): T`
entfernt das angegebene Listenelement. Liefert das entfernte Element zurück bzw. `null`, falls das Element nicht gefunden wurde.

`size(): int`
liefert die Anzahl der Elemente in der Liste zurück.

Material 3

Vorlage UML-Sequenzdiagramm



Material 4

Relationen der Datenbank (Ausschnitt)

Tabelle `Person`

personNr	**nname**	**vname**	**adresse**	**email**
...	...	...	...	...
6641	Vollmers	Hans-Peter	Rodelberg 9, 34537 Bad Wildungen	null
6642	Müller	Sebastian	Hasenwarte 44, 36037 Fulda	sebm.froh@wwweb.eu
6643	Kramer	Kevin	Marktplatz 7, 61462 Königstein	kevin.kramer@kramer-online.eu
6644	Faber	Daria	Großes Moor 5, 36142 Tann	flora123@flora-online.de
6645	Langer	Valentin	Schulstraße 46, 34117 Kassel	route66.vali@t-online.de
6646	May	Peter	Blumenweg 17, 60433 Frankfurt a.M.	null
6647	Siebert	Ariane	Schillerstraße 34, 35708 Haiger	hexe100@fancy-online.eu
...	...	...	...	...

Tabelle `Ferienhaus`

objektNr	**bezeichnung**	**adresse**	**region**	**code**	**land**	**max**	**besitzer**
...	...	...	...	...	...	...	...
100	Haus Teufelsmauer	Osterberg 56, 06502 Thale	1 Harz	DE	Deutschland	5	6644
101	Hütte am Brocken	Brockenstr. 17, 38707 Altenau	1 Harz	DE	Deutschland	3	6644
102	Maison Doro	Korejou, 29880 Plouguerneau	6 Finistiere	FR	Frankreich	6	6645
103	Haus Elsa	Am Weiher 11, 36115 Ehrenberg	2 Rhön	DE	Deutschland	8	6644
104	Villa Lavende	Chemin de la Brize 84200 Carpentras	7 Provence	FR	France	4	6644
...	...	...	...	...	...	...	...

Tabelle `Belegung`

ID	**objektNr**	**belegt_durch**	**von**	**bis**	**typNr**	**typBezeichnung**
1	100	6641	11.03.23	31.03.23	2	Vermietung
2	100	6644	01.04.23	02.04.23	3	Reinigung/Wartung
3	100	6644	23.09.23	07.10.23	4	Eigenbedarf
4	100	6644	01.12.23	28.02.24	1	Gesperrt
5	101	6647	05.06.23	26.06.23	2	Vermietung
6	101	6644	27.06.23	27.06.23	3	Reinigung/Wartung
7	100	6641	10.10.23	24.10.23	2	Vermietung
...	...	...	...	...	...	...

Hinweise: Das Attribut `max` der Tabelle `Ferienhaus` bezeichnet die maximale Personenzahl für das Ferienhaus. Die Attribute `besitzer` der Tabelle `Ferienhaus` und `belegt_durch` der Tabelle `Belegung` referenzieren jeweils das Attribut `personNr` der Tabelle `Person`. Die Primärschlüsselwerte der Tabellen werden automatisch generiert (`AUTO_INCREMENT`).

Material 5

Beispielformular Übergabeprotokoll Ferienhaus

Name/ Anschrift des Mieters: Hans-Peter Vollmers, Rodelberg 9, 34537 Bad Wildungen
Ferienhaus: Haus Teufelsmauer
Adresse: Osterberg 56, 06502 Thale
Datum Einzug: 11.03.2023
Datum Auszug: 31.03.2023

Inventarliste (gekürzt):

Raum	**qm**	**Inventar**					
		Anzahl	**Bezeichnung**	**Beschreibung**	**Kaufdatum**	**Neupreis**	**Zustand**
Wohnraum	25	1	Couch	3-Sitzer	26.03.2015	499.00	gut
		1	Couchtisch	höhenverstellbar	26.03.2015	319.00	gut
		3	Stuhl	Buche massiv	07.03.2022	149.00	neuwertig
		1	Teppich	200x140 cm	11.05.2019	249.90	sehr gut
		1	TV-Gerät TD500	Hersteller: Grundig	16.12.2018	355.99	sehr gut
		...	...	...	...	...	...
Küche	18	1	Esstisch	ausziehbar	12.02.2015	399.99	gut
		6	Stuhl	Stahlgestell	12.02.2015	219.00	gut
		1	Kühlschrank Frosti 100	Hersteller: Boarktis	27.04.2019	399.99	sehr gut
		1	Kochfeld Vulkano77	Hersteller: Boarktis	27.04.2019	239.50	sehr gut
		...	...	...	...	...	...
Schlafraum	21.5	...	...	...	...	...	...
		...	...	...	...	...	...
...	...	...	...	...	...	...	...

☐ keine Mängel festgestellt ☐ Mängel wie folgt festgestellt:

____

Zählerstände:

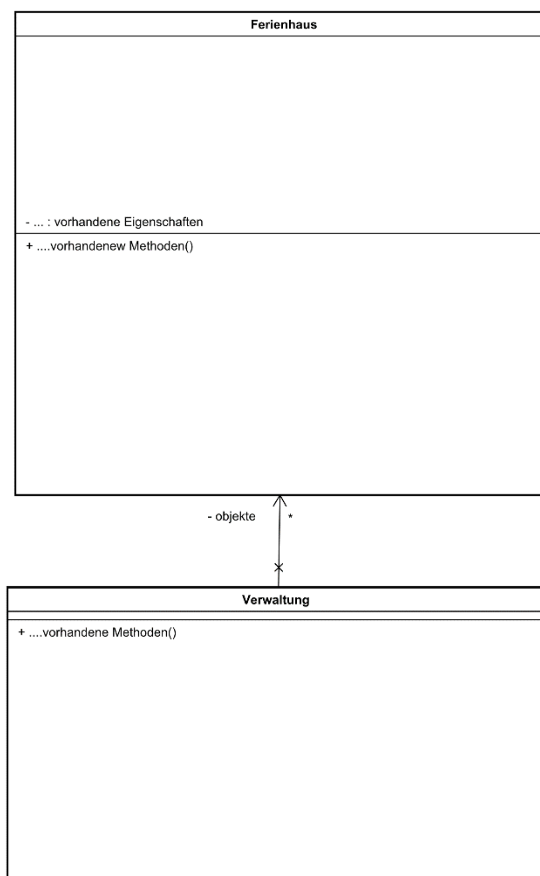
Gas Zählernummer: 11775566 Stand Einzug: 295,255 m³ Stand Auszug: 296,398 m³
Wasser Zählernummer: 66309101 Stand Einzug: 68,217 m³ Stand Auszug: 96,517 m³
Strom Zählernummer: 14620812 Stand Einzug: 14388,1 kWh Stand Auszug: 14512,3 kWh

Schlüssel:

3 Haustür-Schlüssel vorhanden 3 Schlüssel zurückerhalten
1 Garagen-Schlüssel vorhanden 1 Schlüssel zurückerhalten
- Keller-Schlüssel vorhanden Schlüssel zurückerhalten

______________________ ______________________
Ort, Datum, Unterschrift Vermieter Ort, Datum, Unterschrift Mieter

Material 6
Vorlage Erweiterung des Klassendiagramms



Material 7

Vorlage ER-Diagramm

